2015 SMART PROTOCOLS

I. The MeccanoTech™ System

In Fall 2015, Meccano™ released the Meccanoid™ G15 and G15KS robots. These robots are made of plastic parts, which are compatible with traditional Meccano™ parts. The robots also have electronics to control the arms, feet, and head.

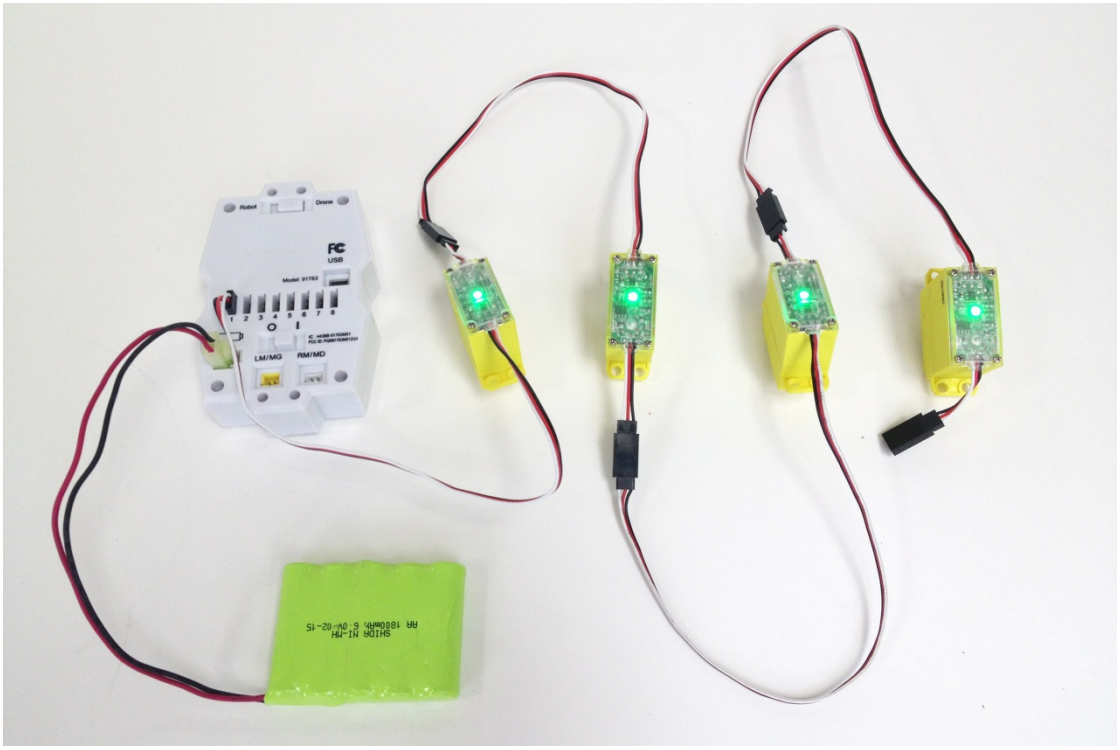
The center of the system is the MeccaBrain™ microcontroller. The arms are made of the plastic parts and the Smart Servo modules. The head is made of plastic parts and the Smart LED module (as well as two Smart Servo modules in the G15KS robot).

II. What is a Smart Module?

In the MeccanoTech™ system, a "smart module" refers to an electronic module which is able to communicate data bi-directionally across a single data line. The Smart Modules all have specific functions. In 2015, Meccano introduced the Smart Servo (for creating rotational motion in the arms and head) and the Smart RGB LED (for creating color in the robot's eyes).

The Smart Modules typically have 1 or 2 sets of three wires: White, Red, and Black. The White wire is the bi-directional data communication. The Red wire is 5 volt power. The Black wire is common Ground.

Because some Smart Modules have 2 sets of wires, they may be "daisy-chained", i.e. one can plug into the other to allow data to be passed along to other Smart Modules. Typically the longer wire is the one upstream of the data.



III. The MeccaBrain™

The MeccaBrain™ is the core microcontroller of the MeccanoTech™ system. It is responsible for controlling all of the Smart Modules.

On the front, the MeccaBrain™ has 4 backlit pushbuttons, a speaker, and a microphone for on-board voice recognition.

The MeccaBrain™ also has Bluetooth Low Energy to communicate with smart phones via the Meccanoid™ app, available for iOS and Android devices.

On the back, the MeccaBrain™ has the main ON/OFF switch and also the ROBOT/DRONE switch. It is recommended to have the MeccaBrain™ switched to ROBOT mode if you are building the Meccanoid™ humanoid. If you are experimenting with any alternate build, you should use DRONE mode. This will prevent the MeccaBrain™ from looking for a specific configuration of Smart modules.

Also on the back, the MeccaBrain™ also has two ports LM and RM for the built-in H-bridges for controlling the DC motors located in the feet. (Note: these are standard motors , not Smart modules.)

There is also the microUSB port. The MeccaBrain™ uses SPI Flash for the memory so its code is updateable via the Robot Updater Software, available for free at http://www.Meccano™.com/Meccanoid™-robot-updater

Lastly, the MeccaBrain™ has eight Smart Module ports: Channels 1 – 8. These 3 pin ports provide Data, Power and Ground to the Smart modules.

While Smart Servos can daisy-chain, there is a limit of 4 modules in a daisy-chain per channel.

In theory, this would mean that you could plug in 32 Smart Servos into the MeccaBrain™ (4 per channel x 8 channels).

However, in reality, the limitation is 16 Smart Servos, due to power constraints.

IV. Smart Servo modules

The Meccano™ Smart Servo is an advanced servomotor capable of bi-directional communication. A servomotor is a motor with built-in angle position feedback so it can rotate to specific angles.

The Smart Servo is not only capable of rotating to specific angles, but it can also disengage the motor and just send the angle position data. This is the basis for Meccanoid™'s "Learned Intelligent Movement" or L.I.M., where the robot will go limp and the user can record an animation into memory by moving the robot's parts.

The Smart Servo also has an RGB LED which is useful as an indicator light. The Smart Servo also has a mechanical clutch and internal current stall detection to protect the Smart Servo from harmful levels of torque.

All Smart Modules are discoverable. This means, that when a new module is plugged in or a module is unplugged, the MeccaBrain™ knows it and can respond.

Most importantly, the Smart Servo has an input plug and an output plug, which allows for a series of Smart modules, or "daisy-chain". Each Smart Module (up to 4 on each channel) passes data along to the next Smart Module. A Smart Module knows its order in the chain, so it is able to determine which data in the packet belongs to itself.

V. Smart LED modules

The Meccano™ Smart LED module in 2015 is an advanced system consisting of two RGB LEDs and is capable of bi-directional communication. RGB means “Red Green Blue” and in theory the LED can create almost any color by combining different levels of light intensity of the three primary colors.

The left RGB LED and right RGB LED are linked so that they each display the same color. Also, the Smart LED module is only has an input plug so it cannot be daisy-chained with other modules. It must the end of the chain.

Unlike the Smart Servo module which only needs a byte from each data packet to function, the Smart LED module requires two data packets to update (1 byte of data from each packet). The reason is that the Smart LED requires 3 bits each of Red, Green and Blue color data and 3 bits of Fade time data.

The color data allows for 2^9 (or 512) color possibilities. The time data allows for 8 levels of preset fade times, the time it takes for one color to transition to another. This is useful so that the user doesn’t have to keep updating the color in steps to transition from one to another seamlessly.

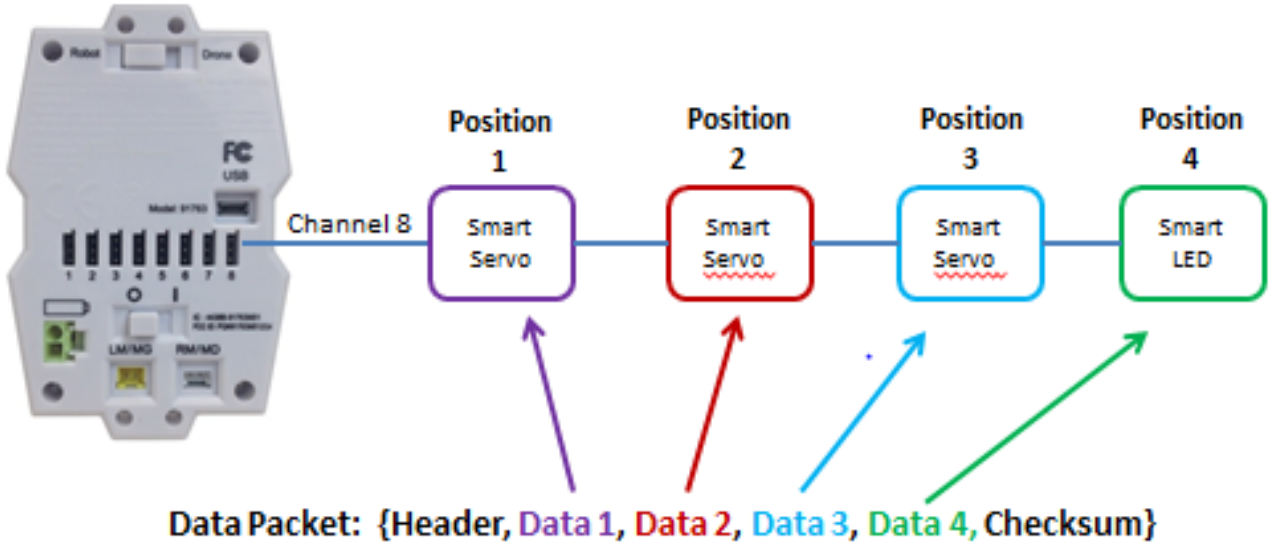
VI. The Data Stream from the MeccaBrain™

Each channel of the MeccaBrain™ sends out its own data stream. The data packets from the MeccaBrain™ consist of 6 bytes: 1 header byte (0xFF), 4 data bytes (one for each theoretical Smart Module) and 1 checksum byte.

The header byte is always 0xFF. Consequently, as a safeguard, no other byte may be 0xFF, neither the data bytes nor the checksum byte.

The data bytes are Data 1, Data 2, Data 3 and Data 4. Data 1 is the data meant for the Smart Module at position 1. Position 1 is the first in the daisy-chain, the one plugged directly into the MeccaBrain™.

Data 2 is the data meant for the Smart Module at position 2. This is the Smart Module plugged into the Smart Module at position 1. Data 3 and Data 4 follow the same pattern. See the example below where 3 Smart Servo modules and a Smart LED module are plugged into Channel 8 of the MeccaBrain™.



The Checksum byte is actually comprised of two 4 bit nibbles, the upper 4 bits and the lower 4 bits. The upper 4 bits are the actual checksum which is calculated based on the values of Data bytes 1 through 4. The lower 4 bits contain the Module ID number which is critical to the entire bi-directional data stream.

Each Smart Module receives the entire 6 byte data packet and then passes the exact same data packet down to the next module. There is no shared data bus. The data going upstream or downstream must be sent *through* each Smart Module.

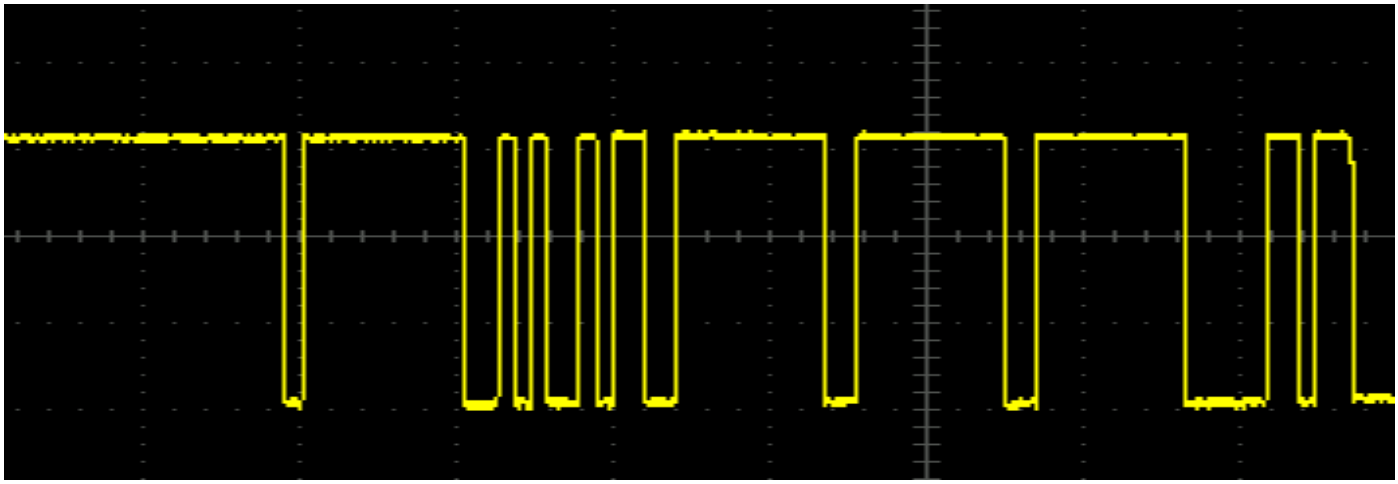
The bits of each byte coming from the MeccaBrain™ are approximately 417us long. A "1" bit is 417us of HIGH; a "0" bit is 417us of LOW. Each byte has a start bit (Low) and two stop bits (High). Also, the data bits are sent in reverse order.

To be clear, each data byte looks like this

{LOW, Bit 0, Bit 1, Bit 2, Bit 3, Bit 4, Bit 5, Bit 6, Bit 7, HIGH, HIGH}

And the entire data packet would look like

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID}



DATA BYTES FROM THE MECCABRAIN TO THE SMART MODULE

{0xFF, 0x4A, 0xFE, 0xFE, 0xFE, 0xB0}

VII. The Data Stream to the MeccaBrain™

The data bytes that are sent from the Smart Modules back to the MeccaBrain™ are sent back in a “round robin” style. Each Smart module takes a turn replying back to the MeccaBrain™’s data packet.

The Module ID from the last byte of the MeccaBrain™’s data packet instructs which Smart Module should reply. There is only one reply for each MeccaBrain™ data packet.

To visualize, see below. BLUE is from the MeccaBrain™. RED is the reply from the Smart Modules

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID = 0}

{Reply Byte from Smart Module 1}

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID = 1}

{Reply Byte from Smart Module 2}

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID = 2}

{Reply Byte from Smart Module 3}

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID = 3}

{Reply Byte from Smart Module 4}

{0xFF, DataByte 1, DataByte 2, DataByte 3, DataByte 4, Checksum/Module ID = 0}

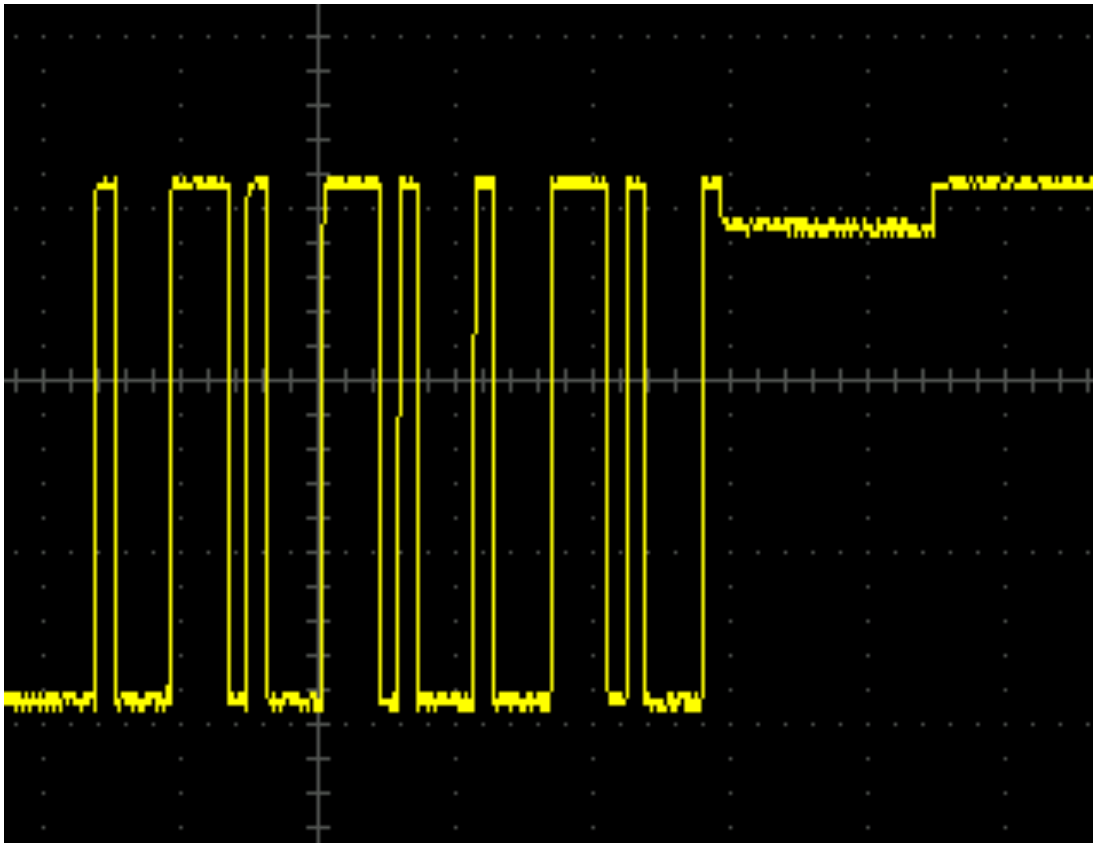
{Reply Byte from Smart Module 1}

AND SO ON

As you can see, the MeccaBrain™ sends data to all of the Smart Modules with each data packet; however, each Smart Module only sends a data packet back on its turn (so a quarter of the time).

The bits of Smart Module data bytes are about 1.1ms long. Each byte has a start bit that is ~2ms LOW. There is no stop bit.

Unlike the bits from the MeccaBrain™ which are just HIGH and LOW for 1 and 0, the bits from the Smart Modules are based on high pulse widths. A "1" bit is when a High pulse is > 400ms; a "0" bit is when the High pulse is < 400ms.



DATA BYTES FROM THE SMART MODULES TO THE MECCABRAIN

{0x52}

VIII. How Smart Modules are discoverable

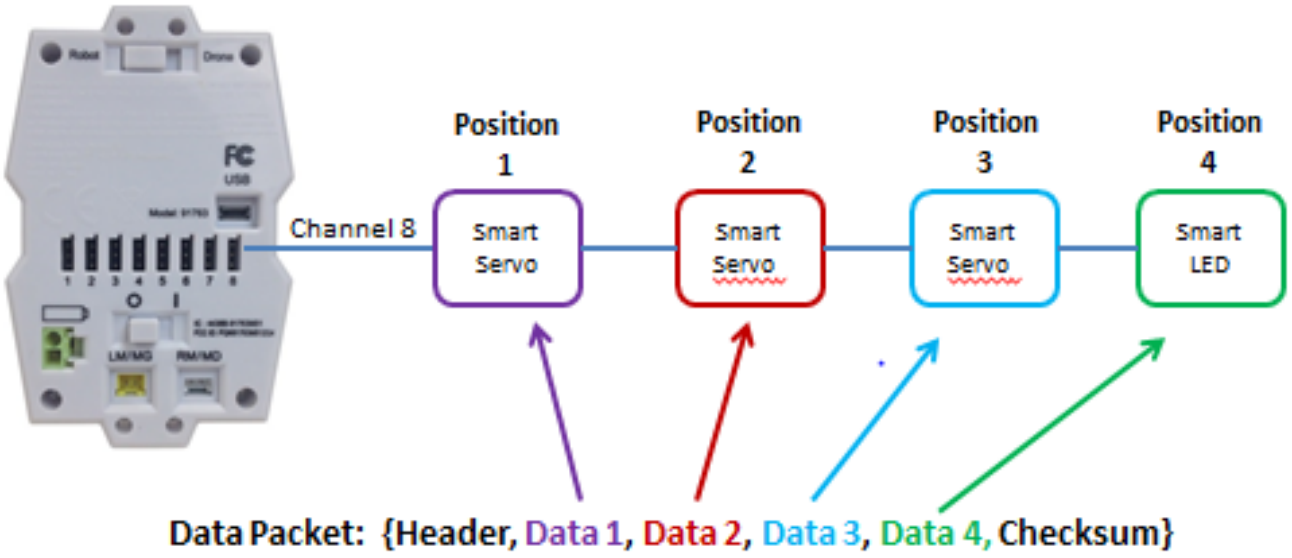
When you plug or unplug a Smart Module, it is recognized by the system, or "discoverable". For Smart Modules, the following bytes are *reserved,* meaning that they are not used for other purposes.

0xFF = header byte (never used for anything else)
0xFE = ID not assigned (means no Smart Module is at this position)
0xFD = Erase ID (resets the Smart Module)
0xFC = Report Smart Module type
0xFB = reserved but not currently used
0xFA – 0x00 = are purely data to the Smart Modules

At startup, the MeccaBrain™ will send out the following data packet

{0xFF, **0xFE**, 0xFE, 0xFE, 0xFE, Checksum/Module ID}

If you plug in a chain of Smart Modules, the first module in the chain will see the first instance of 0xFE in the data packet and realize that it is in Position 1. It will reply back to the MeccaBrain™ with a 0xFE.



The MeccaBrain™ will see the 0xFE reply and realize that there is a Smart Module at Position 1. Then the MeccaBrain™ will change the data byte to 0xFC

{0xFF, **0xFC**, 0xFE, 0xFE, 0xFE, Checksum/Module ID}

The byte 0xFC tells the Smart Module to reply back with its *module type*.

Smart Servos have a module type = 0x01

Smart LEDs have a module type = 0x02

Once the MeccaBrain™ has received the Smart Module's module type, the process is complete. The next bytes that the MeccaBrain™ sends to the module are purely data.

When a module is unplugged, the MeccaBrain™ will be able to recognize that no data is being returned. This is because the Smart Module uses pulse widths of HIGH for 1's and 0's. If the data level is completely LOW, then no data is being returned and the module has been unplugged.

Currently a pull-up resistor circuit is needed when interfacing with the Smart modules (shown below). We are working on ways to improve so that this is not needed in the future.